15. (Amended) The workflow system of claim 12 wherein said means for determining whether a task is eligible for eager execution further comprises:

means for partially evaluating one or more enabling conditions associated with said task.

16. (Amended) The workflow system of claim 12 wherein said means for determining whether a task is eligible for eager execution further comprises:

means for determining whether the task contributes to the production of a target value.

17. (Amended) The workflow system of claim 12 further comprising:

means for determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition depends on one or more outputs of said particular task.

18. (Amended) The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on said particular task.

19. (Amended) The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions depend on one or more outputs of said particular task.

REMARKS

This amendment is submitted in response to the outstanding Office Action, dated November 7, 2003. The present application was filed on February 19, 1999, with claims 1–31. In a previous response to a restriction requirement, Applicants elected to prosecute claims 1–21. Consequently, claims 1–21 are pending. In the outstanding Office Action, the Examiner: (1) rejected claim 3 under 35 USC §112, second paragraph; (2) rejected claims 1 and 12 under 35 USC §102(e); and (3) rejected claims 2–11 and 13–21 under 35 USC §103(a).

With this amendment, Applicants propose to amend the specification and claims 1–8 and 12–19. Applicants respectfully request reconsideration of the outstanding rejections in view of the amendments and the following remarks.

Changes to the Specification

Applicants have amended the specification to correct cross references for related cases, and to correct minor errors of a typographical nature.

Changes to the Claims

With this response, Applicants propose to amend claims 1–8 and 12–19. Claims 1–11 are method claims, and claims 12–21 are apparatus claims. Independent claims 1 and 12 have been amended to change the limitation of "determining whether a task is to be eagerly executed based at least in part on the evaluation of enabling conditions and whether execution of the task results in the initiation of a side-effect action" to the limitation of --determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action--. Additionally, "enabling condition" has been changed to --one or more enabling conditions-- and a limitation of --executing the task using eager execution if the task is determined to be eligible for eager execution-- has been added. These changes are supported, for example, at page 42, lines 18–22 and page 52, line 19 to page 53, line 2 of Applicants' specification. No new matter has been added.

Dependent claims 2–5 and 13–15 have also been amended to comport with the amendments to independent claims 1 and 12.

Claims 6–8 and 17–19 have been amended to provide further clarification. For example, originally filed claim 6 contained the limitation of "determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a task which depends on output [sic] of said particular task." Applicants have amended claim 6 to clarify that it is the enabling condition which depends on the output(s) of the particular task. Similar changes were made to claims 7, 8 and 17–19. These changes are supported, for example, by page 53, lines 3–13 of Applicants' specification. No new matter has been added.

Rejection of Claims

Rejection under 35 USC §112

In the Office Action, the Examiner rejected claim 3 under 35 USC §112, second paragraph, because dependent claim 3 allegedly failed to set forth the subject matter which Applicants regard as their invention. Specifically, the Examiner asserted that claim 3 failed to correspond in scope to the statement in Applicants' specification on page 3, lines 1–3, of "[s]ince non-side-effect tasks have low processing overhead, a non-side-effect task may be eagerly executed even if it is not known whether its associated enabling condition will ultimately be true" (referred to as "the quotation" herein). The Examiner asserted that claim 3 makes "one believe that the enabling condition needs to be known unlike what is stated in the specification where it is not known."

It should be noted that claim 3 and claim 14 have similar limitations. In the following argument, Applicants refer to claim 3, but the argument is also applicable to claim 14.

Original claim 3 has the limitation of "determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that the enabling condition associated with the particular task will evaluate to true." (Note that claim 3 has been amended, but the amended version will not be described here.)

Applicants respectfully submit that claim 3 does correspond in scope with the specification. The quotation in effect states that even if it is not known whether a non-side-effect action's associated enabling condition will ultimately be true, a non-side-effect action may be eagerly executed. Thus, regardless of whether an enabling condition for a non-side-effect action will be true, the non-side-effect action may be eagerly executed. Claim 3 in effect states that prior to determining that the enabling condition associated with the particular task will evaluate to true, there is a determination that a non-side-effect action is eligible for eager execution. Thus, regardless of the evaluation of an enabling condition, a determination may be made as to whether the non-side-effect action is eligible for eager execution. Consequently, in both the quotation and claim 3, the evaluation of the enabling condition is not necessary. (It should be noted that the enabling condition may be partially or completely evaluated, however, as described in detail in Applicants specification.) Therefore both the quotation and claim 3 correspond in scope.

Because both the quotation and claim 3 correspond in scope, Applicants respectfully request that the rejection of claim 3 under 35 USC §112, second paragraph, be withdrawn.

Rejection under 35 USC §102(e)

In the Office Action, the Examiner rejected independent claims 1 and 12 under 35 USC 102(e) as being anticipated by Hoenninger et al., U.S. Patent No. 6,260,058 (hereinafter, Hoenninger). The Examiner asserted that Hoenninger teaches all limitations of independent claims 1 and 12.

Applicants respectfully submit that Hoenninger does not teach or imply all limitations of amended independent claims 1 and 12. Amended independent claims 1 and 12 both contain the limitation of "determining whether a task is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action." Applicants respectfully submit that Hoenninger does not teach or imply this limitation.

Applicants read Hoenninger as allowing a complex control program to be divided into tasks, which are assigned priorities and activation events. See Abstract of Hoenninger. Applicants respectfully submit that Hoenninger does not disclose or imply sub-limitation (1) as it relates to determining whether a task should be eagerly executed, as there is no discussion in Hoenninger of "eager" execution as defined by Applicants or of determining whether a task is eligible for eager execution by consideration of the state of a task. Hoenninger, as pointed out by the Examiner, does describe a status word storing the state of a task (see col. 11, line 24 of Hoenninger) but there is no implication or disclosure in Hoenninger that the state of a task is used to determine whether a task is eligible for eager execution. Furthermore, both sub-limitations (1) and (2) are required for determining whether a task is eligible for eager execution, and Hoenninger does not disclose or imply that either sub-limitation (1) or sub-limitation (2) is used for determination of whether a task is eligible for eager execution.

Concerning sub-limitation (2), in the outstanding Office Action, the Examiner pointed to col. 10, line 66 to col. 11, line 24 of Hoenninger as disclosing sub-limitation (2). As described above, the cited text discloses the structure of a status word for a task, where certain positions in the status word determine what state a task is in. It is unclear to Applicants as to whether any task or subtask in Hoenninger is "a side-effect action" performed external to a workflow system. Nonetheless, even if one were to assume for purposes of argument that Hoenninger discloses a side-effect action, there is no disclosure of the limitation of "determining whether a task is eligible for eager execution by considering at least . . . whether execution of the task results in the initiation of a side-effect action," as claimed by Applicants in amended independent claims 1 and 12.

Applicants respectfully submit that the cited text of Hoenninger apparently describes a status word storing the state of a task. Therefore, Applicants respectfully submit that Hoenninger does not disclose or imply sub-limitation (2) or that sub-limitation (2) is used in a determination of whether a task is eligible for eager execution.

Because Hoenninger does not disclose or imply that either sub-limitation (1) or sub-limitation (2) is used for determination of whether a task is eligible for eager execution, Applicants respectfully submit that amended independent claims 1 and 12 are patentable over Hoenninger.

Rejection under 35 USC §103(a)

In the outstanding Office Action, the Examiner rejected claims 2–11 and 13–21 under 35 USC §103(a) as being unpatentable over Hoenninger in view of one or more other references. Because amended independent claims 1 and 12 are patentable over Hoenninger, dependent claims 2–11 and 13–21 are also patentable, as the dependent claims include all limitations of the independent claims from which they depend.

Moreover, certain of the dependent claims are believed to define additional patentable subject matter. For example, dependent claims 6 and 17 as amended both contain the limitation of "determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition depends on one or more outputs of said particular task." The Examiner, for these claims, combined Hoenninger with Codd et al., U.S. Patent No. 6,421,667 (hereinafter, Codd) and specifically cited col. 16, lines 36–65 and col. 26, lines 22–40 against claims 6 and 17. However, Applicants respectfully submit that the cited text of Codd teaches away from dependent claims 6 and 17.

In particular, Codd states, "In the present embodiment, subsequent task execution only occurs when the resulting truth value is 'TRUE.'" See col. 16, lines 59–61. By contrast, dependent claims 6 and 17 allow one task to be determined as being unneeded even though its enabling condition can be false or never determined. For example, assume that a particular task has a single output that is used in the enabling condition of a second task. If a value of the enabling condition for the second task can be determined without using the single output of the particular task, then the particular task is unneeded, even though the enabling condition for the particular task has never been determined. In Cobb, a value would have to be assigned to the enabling condition for the particular task.

Consequently, Cobb teaches away from dependent claims 6 and 17 and, therefore, these claims are patentable over Hoenninger and Cobb, alone or in combination.

Conclusion

Applicants respectfully submit that the claims as amended herein are patentable over the cited art. If any outstanding issues remain, or if the Examiner has any further suggestions for expediting allowance of this application, the Examiner is invited to contact the undersigned at the telephone number indicated below. The Examiner's attention to this matter is appreciated.

Respectfully submitted,

Dated: January 23, 2003

Robert J. Mauri
Attorney for Applicants
Reg. No. 41,180
Ryan, Mason & Lewis, LLP
1300 Post Road, Suite 205
Fairfield, CT 06430
(203) 255-6560

VERSION WITH MARKINGS TO SHOW CHANGES MADE

IN THE SPECIFICATION

The paragraph beginning at page 1, line 6 has been amended as follows:

This application is related to U.S. patent [application serial] no. 6,424,948 [________ (attorney docket no. Dong 1-4-3-1-3)], entitled Declarative Workflow System Supporting Side-Effects; U.S. patent [application serial] no. 6,499,023 [________ (attorney docket no. Dong 2-7-5-3-2-2-5)], entitled Data Item Evaluation Based on the Combination of Multiple Factors; and U.S. patent application serial no. 09/253,674 [________ (attorney docket no. Hull 6-2-1)], entitled Dynamic Display of Data Item Evaluation; all of which [are being filed concurrently herewith] were filed on February 19, 1999.

The paragraph beginning at page 12, line 8, has been amended as follows:

Each of the modules is associated with an enabling condition, which is a condition which determines whether the module will be evaluated for a given input object. Enabling conditions can refer to attribute values, attribute exception values, attribute states (e.g., whether the attribute has a value or whether an exception occurred when attempting to evaluate it), module states and module exception values. The enabling conditions are graphically represented as broken line hexagons 211, 221, 231, 241, 251, 261. Enabling conditions 211, 251, and 261 contain TRUE, which will always evaluate to a true condition, and therefore the Identify_Caller module 210, Routing_Decisions module 250, and Cal[uc]culate_Wrap_Up module 260 will be evaluated for each input object. Enabling condition 221 contains the expression: VAL (ACCOUNT_NUMBER). The function VAL (X) will return a true condition if the attribute X is in the state VALUE, otherwise, false will be returned. Therefore, the enabling condition 221 indicates that the Info_About_Customer module 220 will be evaluated if the attribute ACCOUNT-NUMBER is in the state VALUE. If the attribute ACCOUNT-NUMBER is in state EXCEPTION, DISABLED, or FAILED, then enabling condition 221 will evaluate to false and the Info_About_Customer module 220 will not be evaluated. If the attribute ACCOUNT_NUMBER is in state UNINITIALIZED, then enabling condition 221 cannot yet be evaluated. Thus, the evaluation of enabling condition 221 depends on the attribute ACCOUNT-NUMBER first receiving a state other than UNINITIALIZED. It is noted that this dependency is implicit in the DL specification and not explicitly specified by the system designer or programmer.

The paragraph beginning at page 13, line 9, has been amended as follows:

As shown in Fig. 4, the module name is specified in line 1, and an indication of which module the current module is a sub-module of is given in line 2. The next section defines the input attributes (line 3). The next section defines the output attributes (lines 4 – 14). Line 15 specifies the enabling condition, which corresponds to the enabling condition 211 shown in Fig. 2. The type of the module, in this case flowchart, is specified on line 16. The computation section of the textual specification (line 17) indicates how attributes are to be evaluated. For this module, the attributes will be evaluated according to the flowchart shown in Fig. 3. Of course, one skilled in the art could convert the flowchart of Fig. 3 to program code to implement the logic flow shown in Fig. 3. Such code is not included in Fig. 4 because it is not necessary for an understanding of the principles of the present invention. Finally, line 18 indicates that this module has a side-effect. The side-effect action is an interactive voice response (IVR) unit [IVR] dip (line 19).

The paragraph beginning at page 15, line 4, has been amended as follows:

The DL specification further defining the Info_About_Customer module 220 (Fig. 2) is shown graphically in Fig. 5 and textually in Fig. 6. This Info_About_Customer module 220 is a declarative module and is therefore further defined in terms of sub-modules. The Get_Recent_Contacts_For_This_Customer module 504, the Get_Recent_Purchases_For_This_Customer module 508, the Get_Account_History_For_This_Customer module 512, and the

Cal[uc]culate_Cust_Value module 528 will always be evaluated because their respective enabling conditions 502, 506, 510, 526 are always true. It is noted that the graphical representation of these modules indicate that they are foreign modules. Each of these modules performs an external database retrieval function. If the attribute RECENT_CONTACTS has a state of VALUE, then the enabling condition 514 will be True and the Calculate_Frustration_Score module 516 will be evaluated. If the attribute RECENT_CONTACTS has state EXCEPTION, DISABLED, or FAILED, then the enabling condition 514 will be False and the Calculate_Frustration_Score module 516 will not be evaluated. If the attribute RECENT_CONTACTS is in state UNINITIALIZED, then enabling condition 514 cannot yet be evaluated. Enabling conditions 518, 522 and 530 are evaluated in a similar manner. The modules 516, 520, 524, 528, and 532 are all represented as solid line hexagons, which indicates that these modules are decision modules and the processing of these modules is specified in terms of computation rules and a combining policy, as will be described in further detail below.

IN THE CLAIMS

l. (Amended) A method for operation of a workflow system for processing an object by executing a plurality of tasks, [each of] one or more of said tasks each having [all] one or more associated enabling [condition] conditions indicating whether the task is to be executed for said object, and wherein execution of at least one of said tasks results in [the] initiation of a side-effect action performed by a component external to said workflow system, said method comprising the [step] steps of:

determining whether a task [is to be eagerly executed based at least in part on the evaluation of enabling conditions] is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action[.]; and

executing the task using eager execution if the task is determined to be eligible for eager execution.

2. (Amended) The method of claim 1 wherein the step of determining whether a task is eligible for eager execution further [comprising] comprises the step of:

determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling [condition] conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

3. (Amended) The method of claim 1 wherein the step of determining whether a task is eligible for eager execution further [comprising] comprises the step of:

determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that the one or more enabling [condition] conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

4. (Amended) The method of claim 1 wherein said step of determining whether a task is [to be eagerly executed] eligible for eager execution further comprises the step of:

partially evaluating [said] one or more enabling conditions associated with said task.

5. (Amended) The method of claim 1 wherein said step of determining whether a task is [to be eagerly executed] eligible for eager execution [is further based on] is performed by also considering (3) whether the task contributes to the production of a target value.

6. (Amended) The method of claim 1 further comprising the step of:

determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition [which] depends on [output] one or more outputs of said particular task.

7. (Amended) The method of claim 1 further comprising the step of:

determining that a particular task is necessary for processing of the object based at least in part on [the] evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions [of tasks that] depend on said particular task.

8. (Amended) The method of claim 1 further comprising the step of:

determining that a particular task is necessary for processing of the object based at least in part on [the] evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions [of tasks that] depend on [the output] one or more outputs of said particular task.

9. (Unchanged) The method of claim 1 wherein said step of determining is performed repeatedly during the processing of the object.

10. (Unchanged) The method of claim 1 wherein a memory of said workflow system stores a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions, said method further comprising the step of:

propagating changes through said graph based on new outputs of completed tasks.

11. (Unchanged) The method of claim 10 wherein said step of propagating changes is based on predefined propagation rules.

12. (Amended) A workflow system for processing an object by executing a plurality of tasks, [each of] one or more of said tasks each having [an] one or more associated enabling [condition] conditions indicating whether the task is to be executed for said the object, and wherein execution of at least one of said tasks results in [the] initiation of a side-effect action performed by a component external to said workflow system, said system comprising:

means for determining whether a task [is to be eagerly executed based at least in part on the evaluation of enabling conditions] is eligible for eager execution by considering at least (1) a state of the task and (2) whether execution of the task results in the initiation of a side-effect action[.]; and

means for executing the task using eager execution if the task is determined to be eligible for eager execution.

13. (Amended) The workflow system of claim 12 wherein the means for determining whether a task is eligible for eager execution further [comprising] comprises:

means for determining that a particular task whose execution results in the initiation of a side-effect action is eligible for eager execution only if it is determined that the one or more enabling [condition] conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

14. (Amended) The workflow system of claim 12 wherein the means for determining whether a task is eligible for eager execution further [comprising] comprises:

means for determining that a particular task whose execution does not result in the initiation of a side-effect action is eligible for eager execution prior to determining that one or more enabling [condition] conditions associated with the particular task will evaluate to true as determined by the state of the particular task.

15. (Amended) The workflow system of claim 12 wherein said means for determining whether a task is [to be eagerly executed] eligible for eager execution further comprises:

means for partially evaluating [said] one or more enabling conditions associated with said task.

16. (Amended) The workflow system of claim 12 wherein said means for determining whether a task is [to be eagerly executed] eligible for eager execution further comprises:

means for determining whether the task contributes to the production of a target value.

17. (Amended) The workflow system of claim 12 further comprising:

means for determining that a particular task is unneeded for processing of the object based at least in part on partial evaluation of an enabling condition of a second task, wherein said second task's enabling condition [which] depends on [output] one or more outputs of said particular task.

18. (Amended) The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on [the] evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions [of tasks that] depend on said particular task.

19. (Amended) The workflow system of claim 12 further comprising:

means for determining that a particular task is necessary for processing of the object based at least in part on [the] evaluation of enabling conditions for a number of tasks, wherein said tasks' enabling conditions [of tasks that] depend on [the output] one or more outputs of said particular task.

20. (Unchanged) The workflow system of claim 12 further comprising:

a memory for storing a graph representing data flow dependencies and enabling flow dependencies between tasks and enabling conditions; and

means for propagating changes through said graph based on new outputs of completed tasks.

21. (Unchanged) The workflow system of claim 20 wherein said memory stores predefined propagation rules and wherein said means for propagating changes further comprises means for propagating changes based on said predefined propagation rules.